Google

P4 HAL for Network Virtualization

Parveen Patel

April 2023

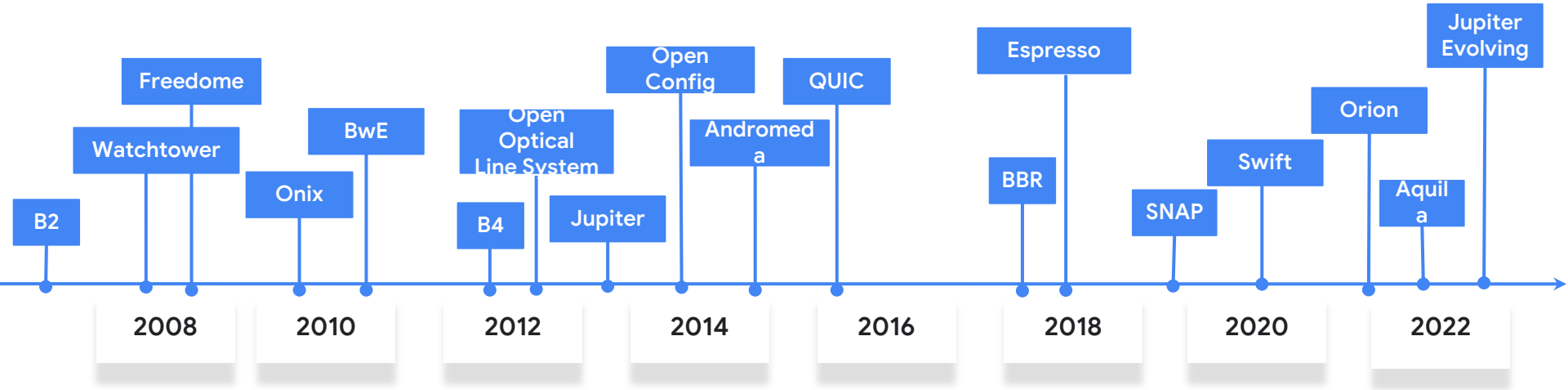
Key takeaways

- **P4 has played a dual role in Switch SDN**
  - As a Hardware Abstraction Layer (HAL) for the Control Plane
  - As an SDK to program the switch pipeline
- **P4 has an even bigger opportunity on SmartNICs**
  - As a HAL for Network Virtualization
  - Enable NIC optionality. Win-win for users and vendors

Outline

1. Google’s SDN journey
2. Lessons from P4 deployment at scale for Switch SDN
3. P4 as a HAL for Network Virtualization
4. Call to action

Google's SDN journey

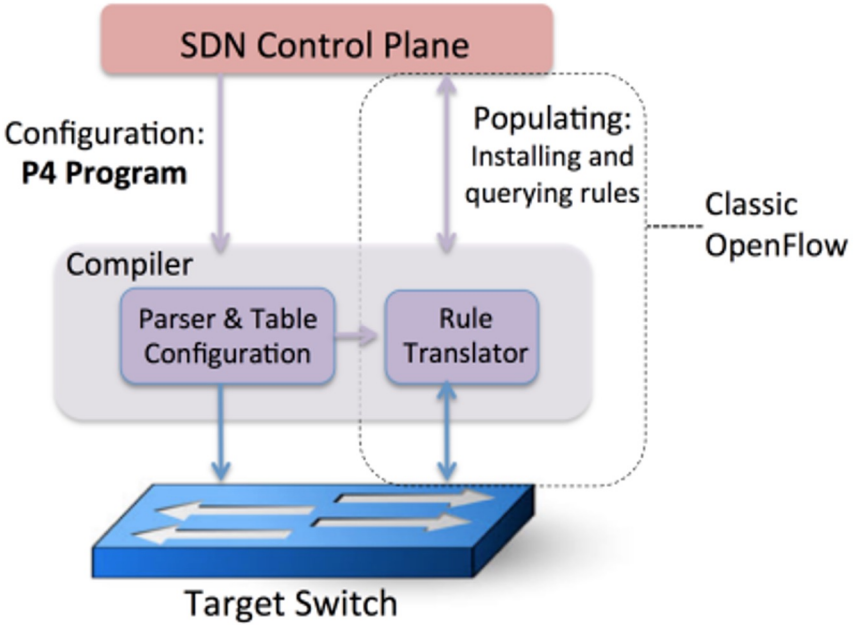


Lessons from P4-based Switch SDN at scale

Google

P4 vision from 2014

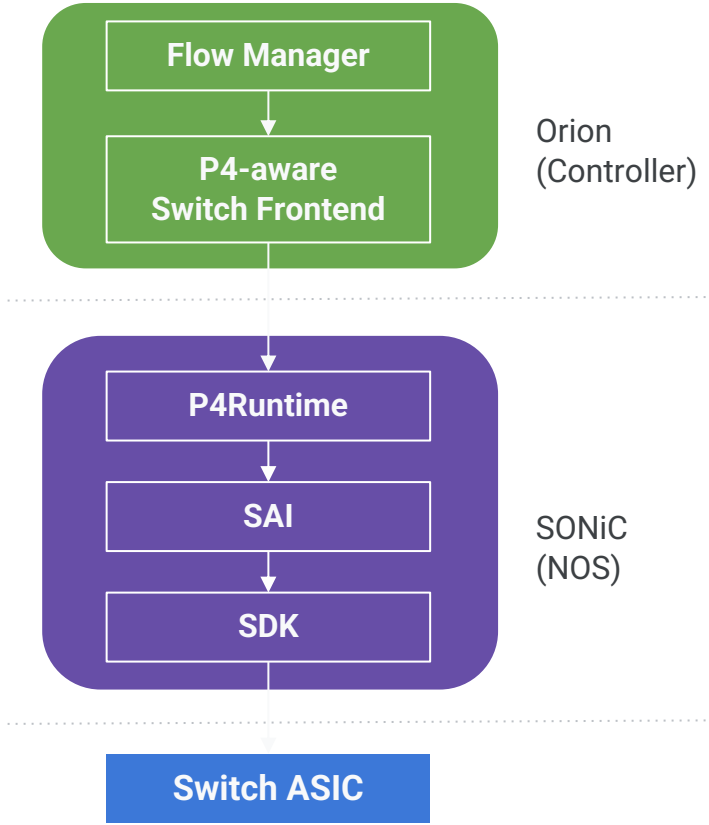
- Language to program packet processors
- Reconfiguration of a switch in the field
- Protocol independence for the switches
- Target independence for the control plane



Ref: http://www.sigcomm.org/sites/default/files/ccr/papers/2014/July/0000000-0000004.pdf

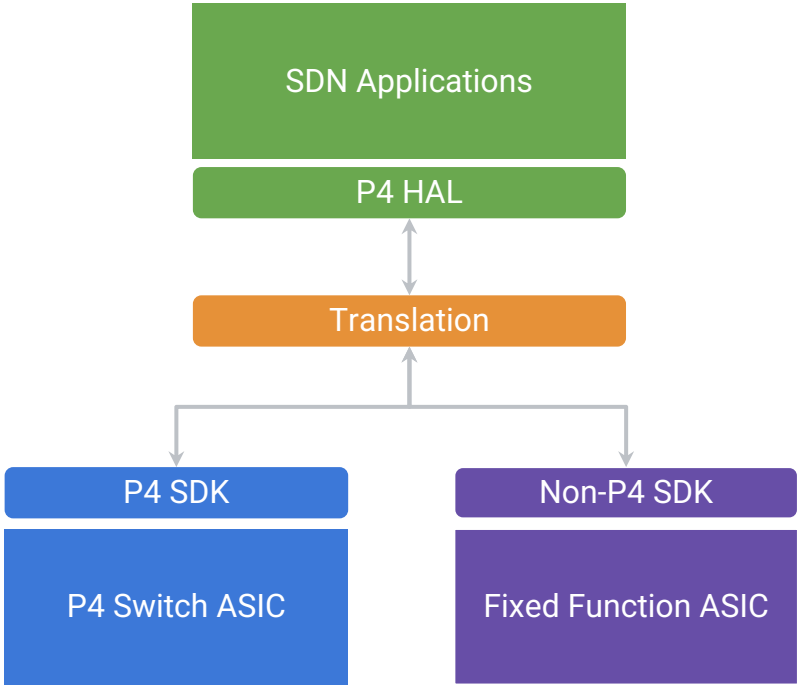
The P4 SDN stack at Google

- Controller's logical view of the switch is in P4
- P4 eventually gets translated to SAI
- SAI is the translation layer between the controller intent and the hardware
- Some switch vendors implement SAI using P4



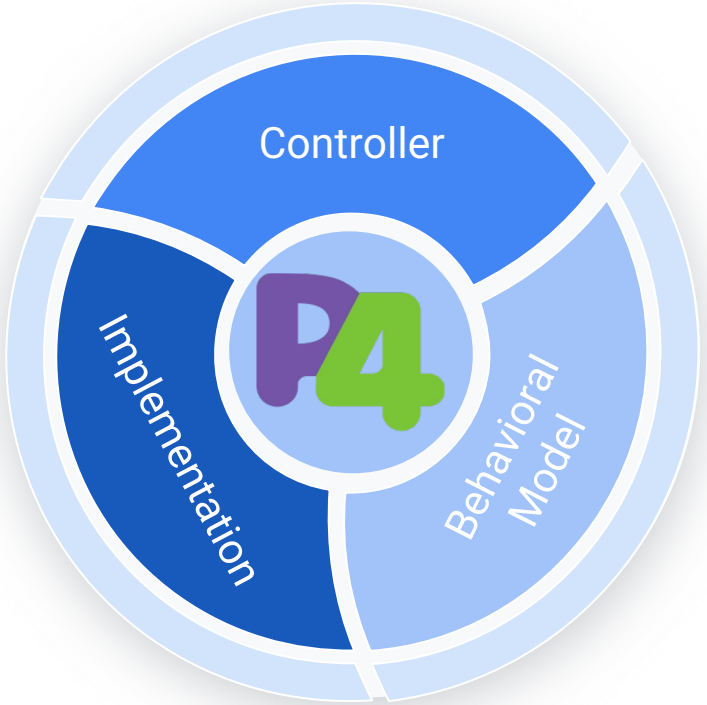
Lessons Learnt: P4 has two related yet distinct use-cases!

- **HAL for SDN applications**
  - Google SDN Controllers
  - Target-independent P4, avoid vendor extensions
- **SDK for programmable hardware**
  - Barefoot, Cisco, AMD
  - Target-specific P4, custom optimization extensions



Lessons Learnt: P4 HAL key benefits

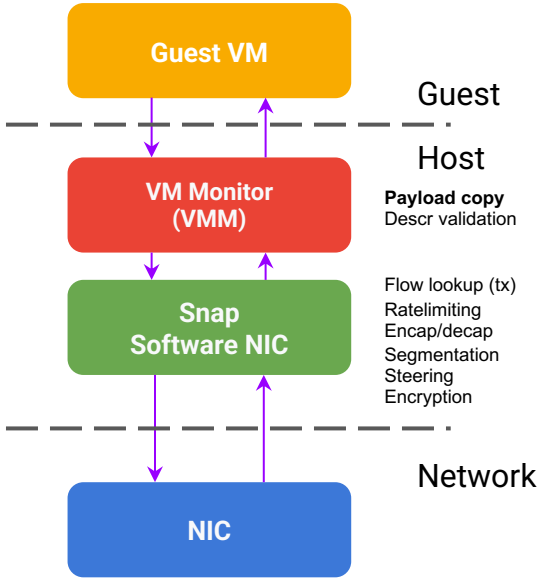
- **Target-independent controller code**
  - Abstraction layer for switch HW
- **Alignment on expected behavior**
  - Formal specification and behavioral model
- **Fewer production bugs**
  - Automated tests



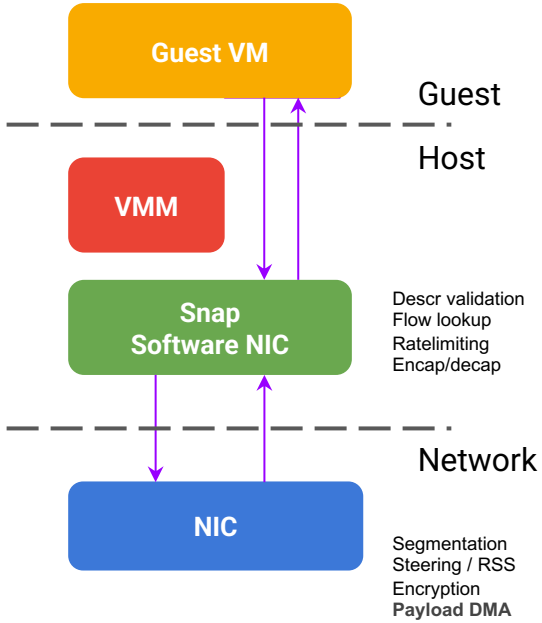
P4 as a HAL for Network Virtualization

Google

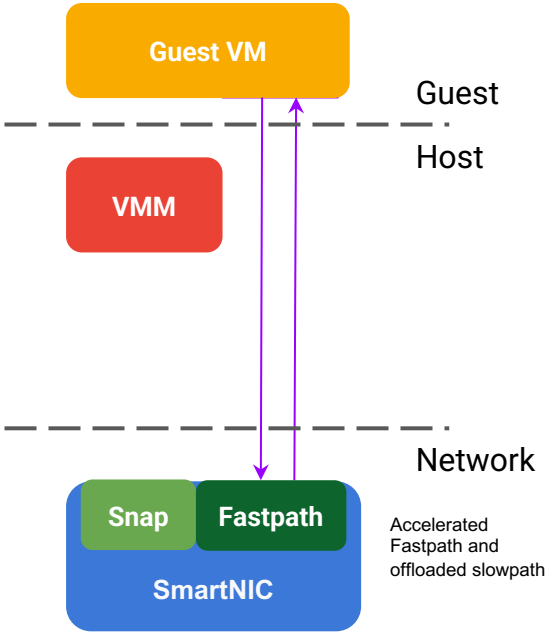
Google's Network Virtualization journey



Andromeda 2.0
2016 - 10 Gbps, minimal offloads



Andromeda 2.2
2019 - 100 Gbps, partial offloads

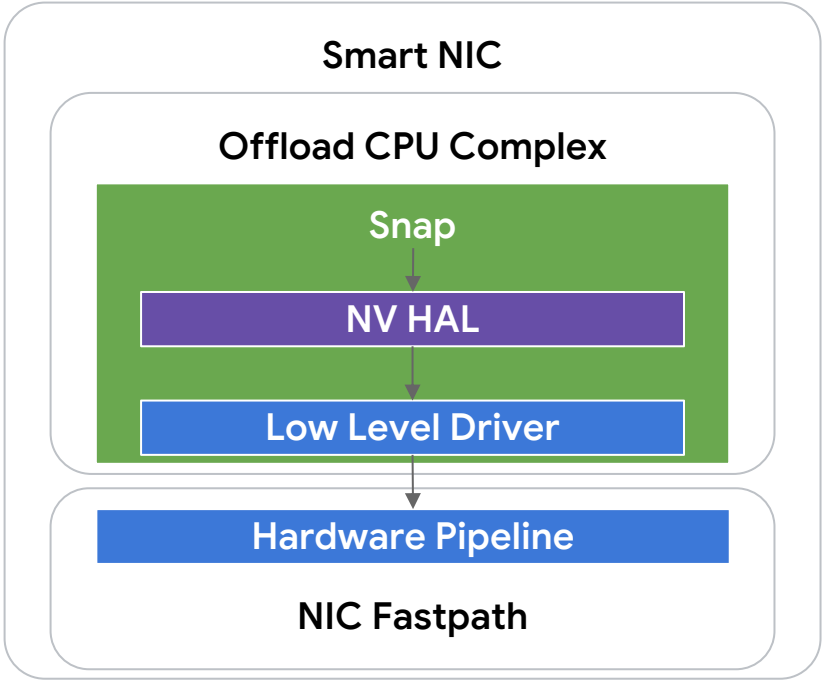


Andromeda 3.0
2022 - 200 Gbps, fully offloaded

Google

Network Virtualization (NV) needs a HAL

- Porting NIC software to different hardware is a multi-year effort
- Need a HAL to enable multiple NICs in the fleet

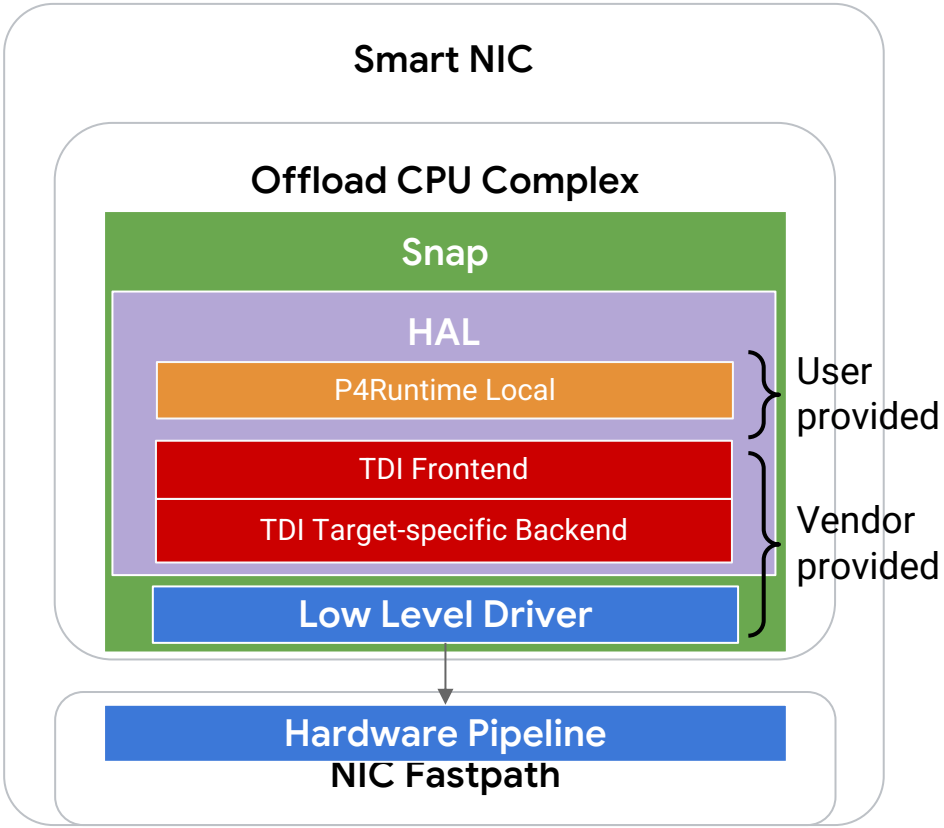


P4 would be a great HAL for Network Virtualization

- P4 has the right building blocks
  - APIs + behavioral model + validation tests
- What's missing?
  - **High-performance P4Runtime**
    - Millions table ops/sec for per-connection insert/delete, bulk ops, counter reads
    - Local controller only, no need for the overhead due to serialization

A sketch of a P4 HAL for Network Virtualization

- **P4 HAL**
  - P4Runtime Local
  - Table Driven Interface (TDI)
- **Provides high performance**
  - Millions table ops/sec
- **Enable a variety of targets**
  - Manual or compiler-driven
- **Enable rapid development of features**
  - No intermediary between HAL and target backend



How to make P4 NV HAL successful?

- Investment in toolchain
  - Especially Testgen and DPDK backend
- User adoption of TDI as an open, vendor-agnostic API
  - For both P4 and non-P4 HALs
- Support for a variety of targets
  - Regardless of the type and degree of programmability

Call to action

- **P4 has played a dual role in Switch SDN**
  - As a Hardware Abstraction Layer (HAL) for the Control Plane
  - As an SDK to program the switch pipeline
- **P4 has an even bigger opportunity on SmartNICs**
  - As a HAL for Network Virtualization
  - Enable NIC optionality. Win-win for users and NIC vendors
- **Call to action**
  - Needs coordinated effort across users and vendors

Backup

P4 addresses a subset of SmartNIC functionality

Much more than a switch-style packet processor

- Multi-modal packet processing: Extensible fastpath, slowpath, transport stacks
- Accelerators: RDMA, compression, cryptography, VM live migration
- Multiple subsystems: NVMe, Hypervisor offloads